1 – الشكل العام لكتابة أي برنامج

البرامج التي تحتوى على دالة Printf أو Scanf **فقط**

```
# include <stdio.h>
Main ( )
{

     هيكل ( جسم البرنامج )

}
```

البرامج التي تحتوى على أي دوال اخرى (Puts - Gets - ....)

```
# include <stdio.h>
# include <conio.h>
Main ( )
{

   هيكل ( جسم البرنامج )
}
```

ملحوظة هامة : كل سطر من اسطر البرنامج لابد أن ينتهي بعلامة الفاصلة المنقوطة (;)
والأوامر التي تكتب داخل البرنامج لابد أن تكون بحروف إنجليزية صغيرة Small

---

2 – حروف لغة C

- الحروف : a – z أو A – z
- الارقام : 0 – 9
- الرموز الخاصة : / - # $ % & > <

---

3 – المعرفات

هي الأسماء التي تعطى لعناصر البرامج المختلفة (المتغيرات) ولها <u>عدة شروط</u> : -

- يمكن أن يحتوى على حروف ابجدية أو ارقام أو علامة الشرطة السفلية (_)
- لا بد أن يبدأ اسم المعرف بحرف أو علامة شرطة سفلية ولا يمكن أن يبدأ برقم

وفى حالة ما اذا كتب اسم المعرف بأحرف يختلف المعرف المكتوب بأحرف كبيرة (A – Z)
عن المعرف المكتوب بأحرف صغيرة (a – z) بمعنى ان المتغير MOH يختلف عن المتغير moh حيث أن الاختلاف بينهم هو حالة الاحرف

- يخلف طول المتغير (المعرف) حسب اختلاف نوع المترجم الخاص بالبرنامج

(Compiler) فهناك مترجم يميز حتى 8 احرف واخر حتى 13 حرف واخر حتى 31 حرف

مثال المترجم الذى يميز حتى 8 احرف لا يستطيع التفرقة بين المتغيرين التاليين

- الاول **file_man**agement
- الثانى **file_man**ager

لان لا يستطيع تمييز الا ثمانية احرف فقط والثمانية احرف متشابهة فى المتغيرين

- لا يجب استخدام معرفات تحمل اسم كلمة محجوزة مثل كلمات جدول صـ 69

هناك امثلة على المعرفات فى جدول صـ 72

4- التعليقات Comments

هى اوامر ايضاحية ليس لها اى تأثير على البرنامج ولا بد ان تبدأ بالرمز * / وتنتهى بالرمز /* مثل التعليق التالى

/ * * Mohammed & Tarek/

---

5 – الثوابت Constants

هناك اربعة انواع من الثوابت : -

- ثوابت صحيحة integer - ثوابت حقيقية (كسرية) (متحركة النقطة) Float
- ثوابت حرفية Character - سلاسل String

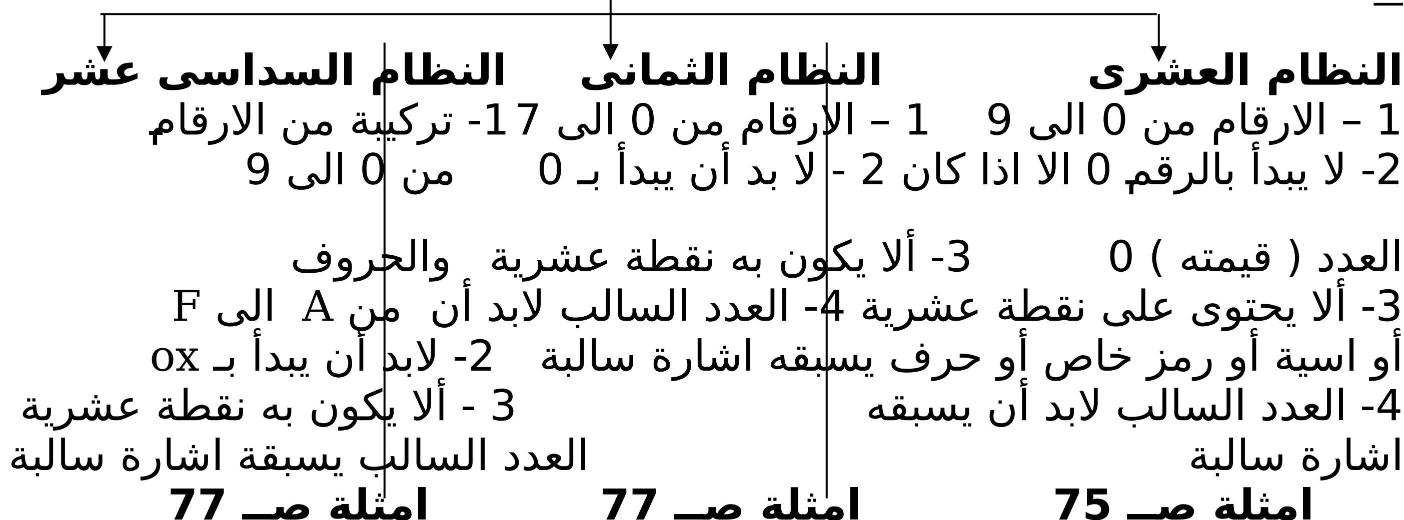
ملاحظات

1- الثوابت الصحيحة والثوابث الحقيقية تمثل اعداد ويشار اليها بأنها ثوابت عددية
2- الثوابت الصحيحة والثوابث الحقيقية لها عدة قواعد
- لا يمكن استخدام الفواصل والفراغات داخلها
- يمكن ان يسبق الثابت اشارة سالبة
- اذا كان الثابت بدون اشارة يجب وضع حرف U أو u فى نهايته
- اذا اردنا التعبير عن قيم اكبر (اطول) للثابت فيجب وضع حرف L أو l فى نهايته
- يمكن وضع الرمز UL أو ul اذا كان الثابت طويل وبدون اشارة
- لا يمكن ان يتعدى الثابت أو يقل عن الحد الاعلى والادنى

اولا : الثوابت الصحيحة (Integer)

هى تلك الاعداد ذات القيم الصحيحة ويمكن ان تكون هذه الارقام موجبة أو سالبة أو صفر

ويمكن ان تكتب بثلاثة انظمة



ثانيا : الثوابت المتحركة (الحقيقية) (Float)

هي الارقام التي تحتوى على علامة عشرية (.) أو اس (e , E) أو كلاهما ويمكن أن يكون العدد موجبا أو سالبا بوضع اشارة (-) قبل العدد

ولها عدة شروط : -

1) يجب أن يحتوى على علامة عشرية أو اس أو كلاهما **مثل** 3.65E6

2) يمكن أن يكون الاس سالب **مثل** 45.6e-10

3) لا يمكن أن تحتوى على حرف أو رمز أو فراغ **مثل** # 4,5 , 235.6

4) لا يمكن أن يكون الاس عشرى **مثل** 9.86E3.6

امثلة صــ 79

ثالثا : الثوابت الحرفية (Character)

هو رمز أو حرف أو رقم واحد أو فراغ يوضع بين فاصلتين (علامة تنصيص فردية)

مثل 'a' , '$' , ' ' , '*' , صــ 83

والثوابت الحرفية لها قيم صحيحة تتحدد طبقا لجدول فئة الرموز ASCII

رابعا : ثوابت السلسلة (String)

ثوابت السلسلة هى التى تحتوى على اى عدد من الرموز أو الفراغات المتتالية محصورة بين علامتى تنصيص مزدوجة " "

مثل "Mohammed @ C . 2009"

---

6 – تسلسلات الهروب (Escape Sequence)

وهو حرف تحكم يبدأ بعلامة الشرطة المائلة (\ ) ثم يتبعها حرف أو رمز تودى وظيفة شكلية معينة

امثلة صـ 85

---

7 – الاعلان عن المتغيرات وتخصيصها (Variable)

هى معرفات أو اسماء رمزية يخصص لها اماكن تخزين فى الذاكرة ولا يوجد قيود على اسماء هذه المتغيرات ولكن لابد من الاعلان عن نوعية المتغيرات بصراحة قبل استخدامها والمتغيرات عدة **انواع هى : -**

المتغيرات الصحيحة Integer	
الاعلان	التخصيص
int a;	a = 150;
الاعلان والتخصيص معا	
int a = 150;	

المتغيرات متحركة النقطة Floating	
الاعلان	التخصيص
float b;	a = 120.58;
الاعلان والتخصيص معا	
float b = 120.58;	

ويلاحظ انه عند تخصيص قيمة للمتغير متحرك النقطة لا بد أن تكون قيمة كسرية (عشرية)

المتغيرات الحرفية Character	
الاعلان	التخصيص
char name;	name = ' moh ';
الاعلان والتخصيص معا	
char name = ' moh ';	

ويلاحظ انه عند تخصيص قيمة للمتغير الحرفى يجب وضعها بين علامتين تنصيص فردية

متغيرات المنظومة Arrays	
الاعلان	التخصيص
char name [5];	name = " moh ";
الاعلان والتخصيص معا	
char name [5] = " moh ";	

ويلاحظ انه عند تخصيص قيمة لمتغير المنظومة يجب وضعها بين علامتين تنصيص مزدوجة

مثال شامل

التفسير	التعبير
الاعلان عن متغير صحيح اسمه num وتخصيص القيمة 55 له	int num = 55;
الاعلان عن متغير سلسلة (منظومة) اسمه name طوله 3 حروف تخصيص القيمة moh لهذا المتغير كما فى السطر الثانى	char name [3]; name = "moh";
الاعلان عن متغير متحرك النقطة اسمه avr وتخصيص القيمة 1.5	float avr = 1.5;
الاعلان عن متغير حرفى اسمه syb وتخصيص الرمز $ له	char syb = '$';

8 – موصفات التحويل Conversion

هي الرموز الدالة على التوصيفات اللازمة للطباعة بمعنى انها عبارة عن رموز توضع نوع عنصر البيانات التي سوف يتم طباعتاه وتبدأ موصفات التحويل بالرمز % ثم يليه رمز واحد أو رمزين أو ثلاثة رموز **مثل %d** فى تعنى أن نوع عنصر البيانات هو رقم صحيح (Integer) **امثلة صــ 99**

موصفات التحويل لها 3 انواع :-

- موصفات تحويل تحتوى على رمز واحد **مثل** **%d**

ولا بد أن يسبق هذا الرمز علامة % ويسمى هذا الرمز (d) رمز التوصيف ويستخدم فى تحديد نوع البيانات وفى هذه الحالة يكون نوع البيانات صحيحة ولتوضح ذلك نفترض المثال التالى

int num = 12345;

تم الاعلان عن متغير صحيح اسمه num وتخصيص القيمة 12345 له فعند طباعة هذه القيمة بأستخدام دالة الاخراج printf تكون الصيغة كما يلى

Printf("%d",num);

الدالة المستخدمة للطباعة — موصف التحويل — اسم المتغير

ونلاحظ انه لا بد من استخدام موصف تحويل حتى يتم توضيح نوع البيانات التي سيتم طباعتها وفى هذه الصيغة موصف التحويل يحتوى على رمز واحد فقط وهو (**d**) وهو يعنى أن نوع البيانات صحيحة ويتم طباعة القيمة

1	**2**	**3**	**4**	**5**

- موصفات تحويل تحتوى على رمزين **مثل** **%3d**

وايضا يسبق هذين الرمزين علامة % الرمز الاول (3) يسمى موصف عرض الحقل ويستخدم فى تحديد الحد الادنى لعرض الحقل وهو ثلاثة حروف والرمز الثانى (d) ويستخدم فى توضيح نوع البيانات التي سيتم طباعتها وهى صحيحة ولتوضيح ذلك ذلك نفترض المثال التالى

int num = 12345;

تم الاعلان عن متغير صحيح اسمه num وتخصيص القيمة 12345 له فعند طباعة هذه القيمة بأستخدام دالة الاخراج printf تكون الصيغة كما يلى

Printf("%3d",num);

الدالة المستخدمة للطباعة — موصف التحويل — اسم المتغير

فى هذه الحالة يكون موصف التحويل مكون من رمزين الاول (3) وهى تمثل الحد الادنى لعرض الحقل والثانى (d) يعنى أن نوع البيانات صحيحة

ويظهر هنا 3 حالات

الحالة الاولى

هي أن القيمة المخصصة للمتغير (خمسة ارقام) > الحد الادنى للحقل (ثلاثة ارقام) فى هذه الحالة يتم طباعة قيمة المتغير بأكملها ويكون الناتج فى هذه الحالة

1	**2**	**3**	**4**	**5**

حفاظا على القيمة الحقيقية للرقم

الحالة الثانية

هي أن القيمة المخصصة للمتغير (خمسة ارقام) < الحد الادنى للحقل (ثمانية ارقام) مثلا أي أن يصبح الموصف %8d فى هذه الحالة يكون عرض الناتج 8 خانات (اماكن) ويتم طباعة الناتج كما يلى

5	4	3	2	1			

نلاحظ الاتى على شكل الناتج

1. يأخذ الناتج ثمانية خانات ويتم البدأ من اليمين الى اليسار بنفس ترتيب الارقام
2. يتم ترك ثلاثة اماكن فارغة وتظهر على شكل مسافة فارغة
3. **اذا كان شكل الموصف %-8d تظهر البيانات بدأ من اليسار الى اليمين وتكون الثلاثة فراغات جهة اليمين كما يلى**

			5	4	3	2	1

4. اذا كان شكل الموصف %08d تظهر البيانات بدأ من جهة اليمين لعدم وجود اشارة –
ولكن يتم تكلمة الفراغات برقم 0 لوجود رقم0 قبل الحد الادنى لعرض الحقل كما بالشكل

5	4	3	2	1	0	0	0

5. لا يمكن أن يكون شكل الموصف %-08d لانه بتطبيق القواعد يتم البدأ من جهة اليسار لوجود اشارة سالبة وبوضع الصفر فى مكان الفراغات أي جهة اليمين هذا يؤدى الى تغيير قيمة الرقم الاصلى من 12345 الى 12345000 **وهذا خطأ**

الحالة	الثالثة

هي أن القيمة المخصصة للمتغير (خمسة ارقام) = الحد الادنى للحقل (خمسة ارقام) مثلا أي أن يصبح الموصف %5d فى هذه الحالة يكون عرض الناتج 5 خانات (اماكن) مثل القيمة المخصصة للمتغير ففى هذه الحالة لايوجد مشكلة وتظهر البيانات كما هي مخصص لها (مثل الحالة الاولى)

5	**4**	**3**	**2**	**1**

- <u>موصفات تحتوى على ثلاثة رموز</u> **مثل** %5.2f

هذا النوع من الموصفات يستخدم فقط مع المتغيرات متحركة النقطة فقط floating حيث انها تحتوى على ثلاثة رموز واحد منها خاص بالارقام العشرية وهى لاتوجد الى فى المتغيرات متحركة النقطة فقط ولتوضيح ذلك نفترض المثال التالى float x = 123.67

تم الاعلان عن متغير متحرك النقطة اسمه x وتخصيص القيمة 123.67 له فعند طباعة هذه القيمة بأستخدام دالة الاخراج printf تكون الصيغة كما يلى

;Printf("%3.2f",num)

الدالة المستخدمة للطباعة — موصف التحويل — اسم المتغير

فى هذه الحالة يكون موصف التحويل مكون من ثلاثة رموز الاول (3) وهى تمثل الحد الادنى لعرض الحقل بالكامل (شامل الكسر العشرى والعلامة العشرية) والثانى (.2) وتستخدم فى تحديد الحد الاقصى لعدد الارقام العشرية التي ستظهر لذلك يكون شكل الناتج عند الطباعة

1	2	3	.	6	7

نلاحظ الاتى على شكل الناتج

1. قاعدة عامة اذا كان حجم المتغير (123.67) وهو سته ارقام شاملة العلامة العشرية اكبر من الحد الادنى لعرض الحقل وهو ثلاثة ارقام فى هذه الحالة يتم وضع قيمة المتغير كاملة حفاظا على القيمة الاصلية للمتغير **اما اذا كان** حجم الارقام العشرية داخل المتغير (.67) وهى رقمين اكبر من الحد الاقصى لعدد الارقام العشرية داخل موصف التحويل فى هذه الحالة يتم تقريب الارقام العشرية حتى يصبح عددها مساويا للحد الاقصى لعدد الارقام العشرية داخل موصف التحويل ولتوضيح ذلك فى المثال السابق نفترض أن الموصف هو %3.1f فى هذه الحالة الحد الادنى لعرض الحقل هو 3 وهو اقل من عدد الارقام داخل قيمة المتغير (123.67) وهى 6 ارقام لذلك يتم كتابة المتغير بأكمله اما الحد الاقصى لعدد الارقام العشرية هو .1 أي رقم واحد وهو اقل من عدد الارقام العشرية فى المتغير وهى (.67) وهى رقمين لذلك يتم تقريب الرقمين الى رقم واحد فقط ويكون .7 بتطبيق قواعد التقريب الرياضية ويكون شكل الطباعة كما يلى بالنسبة للتوصيف %3.1f

1	2	3	.	7

2. اذا كان التوصيف هو %10.4f للمتغير الذى قيمته 123.67 فى هذه الحالة يكون الحد الادنى لعرض هو 10 ارقام وهو اكبر عدد الارقام داخل المتغير وهى 6 ارقام فى هذه الحالة يظهر فراغين جهة اليسار لان البدأ سيكون من جهة اليمين والحد الاقصى للارقام العشرية هو 4 ارقام اكبر من عدد الارقام العشرية داخل المتغير وهى 2 رقم لذلك يتم تكملة الفرق فى العلامة العشرية بوضع اصفارويظهر الناتج كما يلى

		1	2	3	.	6	7	0	0

3. اذا كان التوصيف هو %-10.4f سوف يظهر الفراغين ناحية اليمين

4. اذا لم يقوم بتحديد الحد الاقصى لعدد الارقام العشرية فى هذه الحالة يتم كتابة كافة الارقام العشرية الموجوده داخل المتغير ولابد أن يكون الارقام العشرية حتى 6 ارقام فإذا كان عدد الارقام العشرية داخل المتغير رقمين فقط يتم اضافة 4 اصفار بجانبها لاتمامها حتى 6 ارقام عشرية ولتوضيح ذلك نفترض التوصيف التالى %4f للمتغير الذى قيمته 123.67 يكون شكل الناتج عند الطباعة كما يلى

1	2	3	.	6	7	0	0	0	0

واذا كان التوصيف %13f يكون الناتج عند الطباعة

			1	2	3	.	6	7	0	0	0	0

واذا كان التوصيف %-13f يكون الناتج عند الطباعة

1	2	3	.	6	7	0	0	0	0			

واذا كان التوصيف %4.4f يكون الناتج عند الطباعة

1	2	3	.	6	7	0	0

يصبح عرض الحقل 8 حيث انه هناك 4 ارقام عشرية و3 ارقام صحيحة والعلامة العشرية

<u>مثال عام</u>

اكتب شكل المخرجات فى حالات التوصيف التالية بفرض المتغيرين التاليين

:Int x =12345
float y = 123.55

التوصيف	1	2	3	4	5	6	7	8	9	10	11	12
printf("%d",x)	1	2	3	4	5							
printf("%3d",x)	1	2	3	4	5							
printf("%8d",x)				1	2	3	4	5				
printf("%-8d",x)	1	2	3	4	5							
printf("%08d",x)	0	0	0	1	2	3	4	5				
printf("%-08d",x)	غير ممكن											
printf("%f",y)	1	2	3	.	5	5	0	0	0	0		
printf("%-f",y)	1	2	3	.	5	5	0	0	0	0		
printf("%8f",y)	1	2	3	.	5	5	0	0	0	0		
printf("%12f",y)			1	2	3	.	5	5	0	0	0	0

printf("%-12f",y)	1	2	3	.	5	5	0	0	0	0		
printf("%8.4f",y)	1	2	3	.	5	5	0	0				
printf("%-8.4f",y)	1	2	3	.	5	5	0	0				
printf("%8.1f",y)				1	2	3	.	6				
printf("%-8.1f",y)	1	2	3	.	6							
printf("%2.1f",y)	1	2	3	.	6							
printf("%-.2f",y)	1	2	3	.	5	5						
printf("%.4f",y)	1	2	3	.	5	5	0	0				
printf("%10.4f",y)			1	2	3	.	5	5	0	0		
printf("%-10.4f",y)	1	2	3	.	5	5	0	0				

تم بحمد الله

مع تمنياتى بالنجاح والتوفيق

محمد السيد إسماعيل

1 – المؤثرات الحسابية (العمليات الحسابية)

وهى + , - , * , / , %

الجمع , الطرح , الضرب , القسمة , باقى القسمة

ملاحظات

3- هناك اولويات للعمليات الحسابية وهى كما يلى

- فك الاقواس من الداخل الى الخارج
- الاسس
- الضرب والقسمة وباقى القسمة من اليسار الى اليمين
- الجمع والطرح من اليسار الى اليمين

4- يجب مراعاة قاعدة الترقية

- اذا كان المتغيرات من نفس النوع فإن الناتج يكون من نفس النوع مثال اذا تم
  جمع متغيرات صحيحة فإن الناتج يكون صحيح
- اذا كان المتغيرات من انواع مختلفة فيكون الناتج من النوع الاكثر دقة مثال اذا
  تم جمع متغير صحيح مع متغير حقيقى (متحرك) يكون الناتج حقيقى وهذا ما
  يسمى بالترقية واذا تم جمع متغير حرفى مع متغير صحيح يكون الناتج صحيح
  وايضا يتم ترقية المتغير الصحيح الى (Long)

3- **<u>بالنسبة لموثر باقى القسمة (%)</u>**

- يجب ان تكون قيم المتغيرات صحيحة وإلا فالناتج يكون خطأ مثل 5%2 فيكون
  الناتج = 1 اما 5.4%2 لايمكن اجرائها لان البسط متغير حقيقى
- اشارة الناتج تكون نفس اشارة العامل الايسر (البسط) مثل -5%2 فيكون
  الناتج = -1 اما 5%-2 فيكون الناتج = 1 لان البسط موجب

4- **<u>بالنسبة لموثر القسمة (/)</u>**

اذا كان البسط اقل من المقام مثل 2/5 يكون الناتج = 0 وباقى القسمة = 2

<u>2 – المؤثرات الاحادية</u>

وهى - , ++ , --

- الموثر (-) يدل على اشارة المتغير أي أن قيمته سالبة
- المؤثر (++) يؤدى الى زيادة قيمة المتغير بمقدار 1
- الموثر (--) يؤدى الى انقاص قيمة المتغير بمقدار 1

ملحوظة

يوثر موقع المؤثر من المتغير على قيمة المتغير

- اذا كان الموثر يسبق المتغير مثل ++4 فإن قيمة المتغير الجديدة = 5
- اذا كان المؤثر يلى المتغير مثل 4++ فإن قيمة المتغير = 4 ولكن اذا ظهر فى
  عملية اخرى فى نفس البرنامج تلى هذه العملية تصبح قيمته = 5

مثال عملى 1

r = 3 , q = 3 , P = 3 احسب ناتج العملية التالية

P++/2+--q*++r	= 1+8 = 4*2 + 3/2 = **9**
P++/2+--q*++r+P	**13** = 4 + 4*2 + 3/2 = ويلاحظ أن قيمة p الاخيرة = 4

---

3 – المؤثرات العلاقية

وهى == , > , < , =< , =>

ويكون ناتج المقارنة

- 0 ⟶ False
- 1 ⟶ True

---

4 – المؤثرات المنطقية

وهى && , || , !

Not , Or , And

وتستخدم فى المقارنة بين تعبيرين

	التعبير الاول	التعبير الثانى	الناتج
&&	True	True	True
	اى شئ خـــلاف ذلك		False
||	False	False	False
	اى شئ خـــلاف ذلك		True

- **!** تعكس الـ True الـى False والعكس

مثال عملى 2

c = 119 , f = 5.5 , i = 7 احسب ناتج العملية التالية

التعبير	التفسير	القيمة
(c==119) || (i>=6)	True || True	1
(10=>(i+f))||(c!=11)	True || False	1

---

5 – مؤثرات التحديد

تستخدم لعمل تعبيرات تحديد تحدد فيها تعبير لاحد المعرفات (المتغيرات) وهى ثلاثة

متغير = متغير **مثل** x = **y**

ثابت = متغير **مثل** x = 5

تعبير = متغير **مثل** **x = y + 5**

فنلاحظ دائما : أن الطرف الايسر يكون متغير والطرف الايمن اما أن يكون متغير أو ثابت أو تعبير ولا يمكن تخصيص تعبير لتعبير **مثل** x + y = length * width

ملحوظة

- فى حالة الاعلان عن متغير صحيح وتخصيص قيمة حقيقية له فإن هذه القيمة تتحول الى قيمة صحيحة بإستبعاد الجزء العشرى **مثل** 5.63 تتحول الى 5
- يمكن تخصيص قيم لاكثر من متغير **مثل** m = g = 1
- هناك اختلاف بين مؤثر التحديد (=) ومؤثر التساوى (==)

6 – مؤثرات التحديد الحسابية المركبة

مثل += , -= , *= , /= , %=

مثال **x = x + y** ⟶ **x += y**

x = x%y ⟶ **x %= y**

7 – المؤثرات الشرطية

الصيغة العامة لها هى

(الشرط)	?	القيمة فى حالة تحقق الشرط	:	القيمة فى حالة عدم تحقق الشرط

ملحوظة

- لابد أن يوضع الشرط داخل اقواس
- لابد أن توضع علامة استفهام (?) قبل القيمة فى حالة تحقق الشرط
- لابد أن توضع علامة نقطتين (:) قبل القيمة فى حالة عدم تحقق الشرط

مثال عملى 3

i = 3 , j = 4 , k = 5 احسب ناتج العملية التالية

k+=(i>0&&i<=15)?++i:i/j

الحل

k+=(True && True) ? 4 : 0

k+=4

k=k+4

k=9 k=5+4

بفرض أن

i = 20 , j = 10 , k = 30

k+=(True && False) ? 21 : 2

k+= 2

k=k+2 k=30+2 k=32

دوال الادخال و دوال الاخراج

1 – دالة الاخراج العامة (Printf ( ))

تستخدم فى اخراج البيانات العددية أو النصية أو السلاسل **والصيغة العامة لها هي :-**

اولا : البيانات العددية

وهى اما أن تكون متغيرات صحيحة أو متحركة أو ارقام

;printf ("format" , var1 , var2, ...)

اسم الدالة — توصيف الطباعة — المتغيرات

مثال

;printf("%d",x+y)

(حيث أن **y , x** متغيرات صحيحة **)**

ثانيا : البيانات النصية

وهى تشمل النصوص المراد اخراجها

;printf ("Text")

مثال

;printf("Mohammed:0124828233")

ثانيا : البيانات نصية وعددية معاً

حيث انه يمكن اخراج بيانات نصية وعددية معاً

مثل

;printf("x = %d y = %f " ,x,y)

وبفرض أن **x = 5 , y = 5.62**

اذاً يكون شكل الناتج كما يلى

x = 5 , y = 5.62

حيث انه اعتبر (**x =**) نص فيظهر كما هو ويقوم المترجة بطباعة قيمة بعد علامة (=) بتوصيفة على انه صحيح (d%) وهكذا مع (**y**)

ملاحظات

- لابد أن يوضع التوصيف (d%) بين علامتى تنصيص مزدوجة (" التوصيف ")
- لابد من وصع علامة الفاصلة (,) قبل المتغير الاول و ايضا بين المتغيرات
- لابد ان ينتهى السطر بعلامة الفاصلة المنقوطة (;)
- يمكن أن يحتوى الجزء الخاص بتوصيف الطباعة (format) الموجود داخل دالة الاخراج على تسلسلات الهروب **مثل** printf("\n %d",x)

حيث تقوم سلسلة الهروب (\n) بطباعة الناتج على سطر جديد

بعض تسلاسلات الهروب

n\	تعنى الانتقال الى سطر جديد
a\	تعنى اصدار صوت جرس
r\	تعنى البدء من اول السطر
t\	تعنى تقدم 7 مسافات عمودية قبل الطباعة

امثلة عملية على دالة الاخراج printf

<u>اعرض شكل الناتج الذى سيتم طباعتة من البرامج التالية</u>

<u>مثال 1</u>

```
;Int c=3
;c =++c+c
;printf("The Value Of  c = %d",c)
```

<u>ناتج الطباعة</u>

The Value Of c = 8

<u>مثال 2</u>

```
;Int c=3
;printf("The Value Of  c = %d",++c+c)
```

<u>ناتج الطباعة</u>

The Value Of c = 8

<u>نلاحظ ان</u> المثالين لهم نفس الناتج ولكن الاختلاف هو ان العملية الحسابية فى البرنامج الاول فى سطر داخل البرنامج اما العملية الحسابية فى البرنامج الثانى تكون داخل دالة printf

<u>مثال 3</u>

```
;'Char x= 'a
;printf("\n x= %c",x)
```

ناتج الطباعة

x= a

مثال 4

```
;printf("%d ",100+10)
;printf("%f",100.0+10)
```

ناتج الطباعة

110
110.000000

فى عملية الطباعة الاولى المتغير صحيح وتوصيفة ايضا صحيح (d%) فبالتالى يكون الناتج صحيح (110) وبعد طباعة السطر الاول يتم الانتقال الى سطر جديد حيث انه يوجد تسلسل هروب (n\ ) للانتقال الى سطر جديد ثم طباعة السطر الثانى

فى عملية الطباعة الثانية المتغيرات احدها صحيح والاخر متحرك وطبقا لقاعدة الترقية يكون الناتج متحرك والتوصيف ايضا متحرك (f%) ويتم اضافة 5 اصفار عشرية

مثال 5

```
;printf("\n%d\n",140.5*2)
;printf("%f\n",140.5*2)
;printf("%f",140*2)
```

ناتج الطباعة

0
281.000000
0.000000

نلاحظ ان

- فى عملية الطباعة الاولى تم توصيف المتغيرات على انها صحيحة (% d) ولكن هى حقيقية لذلك يكون ناتج الطباعة خطأ ويظهر هذا الخطأ فى شكل صفر صحيح (0)
  ويمكن معالجة هذا الخطأ بالصيغة التالية printf("\n%d\n",int ((140.5*2);
  ففى هذه الحالة يكون الناتج (281)
- فى عملية الطباعة الثانية تم توصيف المتغيرات على انها حقيقية (% f) وهى بالفعل حقيقية لذلك يكون ناتج الطباعة حقيقى (281.000000) ويجب مراعاة موصف عرض الحقل وموصف الارقام العشرية ان وجد **مثل** (3.2f%) فيصبح الناتج (281.00)

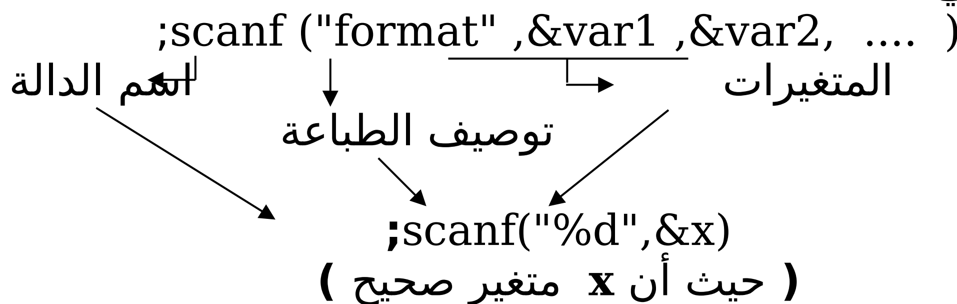
- فى عملية الطباعة الثالثة تم توصيف المتغيرات على انها حقيقية (%f) ولكن هى صحيحة لذلك يكون ناتج الطباعة خطأ ويظهر هذا الخطأ فى شكل صفر متحرك (0.000000) **مثال صـ 176 (هام جدا)**

---

2 – دالة الادخال العامة (scanf ( ))

تستخدم فى ادخال البيانات العددية أو النصية أو السلاسل **والصيغة العامة لها هي :-**

scanf ("format" ,&var1 ,&var2, ...);



مثال

scanf("%d",&x)**;**

(حيث أن **x** متغير صحيح **)**

ملاحظات

- لابد أن يوضع التوصيف (%d) بين علامتى تنصيص مزدوجة (" التوصيف ")
- لابد من وصع العلامة (&,) قبل المتغير الاول و ايضا بين المتغيرات
- لابد ان ينتهى السطر بعلامة الفاصلة المنقوطة (;)
- يمكن أن يحتوى الجزء الخاص بتوصيف الطباعة (format) الموجود داخل دالة الاخراج على تسلسلات الهروب **مثل** scanf("\n %d",&x)
- اسماء المنظومات الحرفية يجب ألا يبدأ بالرمز (&) مثل scanf("%s",name)

امثلة عملية على دالة الادخال scanf

اعرض شكل الناتج الذى سيتم طباعة من البرامج التالية

مثال 1

```
Int I ;
char a,b;
scanf("%c %c %d \n" ,&a ,&b ,&I);
printf(" A = %c B = %c  I = %d " ,a ,b ,I);
```

اولا: يطلب البرنامج ادخال قيم للمتغير a , b , I واذا تم ادخال (mg1)

ثانيا : يظهر الناتج A= m B =g I = 1

مثال 2

```
float a;
```

```
scanf("%f " ,&a );
printf(" a = %.2f " ,a);
```

الناتج

اولا: يطلب البرنامج ادخال قيم للمتغير a واذا تم ادخال (2501.268533)

ثانيا : يظهر الناتج (2501.27)

مثال 3

(هام جدا) صـ 183

---

3 – دوال الادخال الخاصة (getchar ( ) , getche ( ) , getch ( ))

تستخدم جميعها لادخال حرف واحد فقط من لوحة المفاتيح

اولا : دالة getch ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح دون الضغط على زر Enter من لوحة المفاتيح وهذا الحرف لايظهر على الشاشة عند ادخاله

ثانيا : دالة getche ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح دون الضغط على زر Enter من لوحة المفاتيح وهذا الحرف يظهر على الشاشة عند ادخاله

ثالثا : دالة getchar ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح ولا يتم الانتقال الى الامر التالى إلا اذا تم الضغط على زر Enter من لوحة المفاتيح وهذا الحرف يظهر على الشاشة عند ادخاله

مثال 1

```
char a;
a = getche ( );
printf(" %c" ,a);
```

الدالة getche تودى الى ظهور الحرف a دون الضغط على enter لذلك فيظهر الحرف مرتين

4 – دوال الاخراج الخاصة (putch ( ))

تستخدم لادخال حرف فقط على الشاشة

```
char ch1, ch2, ch3;
printf ("\n ch1= ");
ch1=getchar ( );
printf ("\n ch2= ");
ch2=getche ( );
printf ("\n ch3= ");
ch3=getch ( );
putch (ch1);  putch (ch2);   putch (ch3);
```

- فى السطر الثانى يطبع ch1= على سطر جديد
- فى السطر الثالث يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف m فيظهر هذا الحرف ولابد من الضغط على مفتاح enter للانتقال الى السطر التالى
- فى السطر الرابع يطبع ch2= على سطر جديد
- فى السطر الخامس يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف g فيظهر هذا الحرف ولا يتم الضغط على مفتاح enter حيث ينتقل تلقائيا الى السطر التالى
- فى السطر السادس يطبع ch3= على سطر جديد
- فى السطر السابع يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف s دون أن يظهر هذا الحرف ولا يتم الضغط على مفتاح enter حيث ينتقل تلقائيا الى السطر التالى
- فى السطر الثامن يطبع (m g s)

مثال صــ 188

---

5 – دالة الاخراج (puts ( ))

تستخدم لطباعة عبارة أو منظومة حرفية فى نفس صورة ادخالها بدلاً من printf بالنسبة للمنظومات الحرفية

6 – دالة الادخال (gets ( ))

تستخدم لادخال عبارة أو منظومة حرفية فى نفس صورة ادخالها بدلاً من scanf بالنسبة للمنظومات الحرفية

ملاحظات

- دالة puts ( ) ودالة gets ( ) تستخدم للمنظومات والعبارات الحرفية فقط
- ودالة gets ( ) تستقيل البيانات الحرفية بكافة انواعها بما فيها الفراغات

مثل عند ادخال كلمة (Mohammed Ismeal) تطبع كما هي بالفراغات التي بداخلها

- دالة scanf( ) لاتسمح بوجود مسافات داخل العبارة وتعتبرها فاصلة وتأخذ اول مقطع فقط

مثل عند ادخال كلمة (Mohammed Ismeal) تطبع المقطع الاول فقط Mohammed

---

مثال 1

```
char line [200];
gets (line);
puts (line);
```

الناتج

- فى السطر الثانى يطلب البرنامج ادخال منظومة حرفية تسمى line ومكونة من 200 حرف فاذا قام المستخدم بأدخال

(When we take ..we may get little.. but when God gives… HE gives us more beyond our expectations)

- فى السطر الثالث يقوم البرانامج بطباعة العبارة فى نفس صورة ادخالها

---

7 – دالة مسح الشاشة (clrscr ( ))

تستخدم لمسح الشاشة ووضع المؤشر فى اول الركن العلوى الايسر من الشاشة

8 – دالة تغير لون الكتابة (textcolor ( ))

تستخدم لتغير لون الكتابة والصيغة العامة لها هي

textcolor (color no)

حيث يتم كتابة رقم اللون وهو من الرقم 0 الى الرقم 14

أو تكون الصيغة العامة لها كما يلى

textcolor (color name)

حيث يتم كتابة اسم اللون مثل اللون الاحمر (RED)

ملاحظات

- لطباعة نص باللون المحدد تستبدل دالة printf بدالة cprintf
- لطباعة اكثر من سطر كل سطر بلون (مثل حلقة) تستبدل سلسلة الهروب \n بـ \r\n

8 – دالة تغير لون خلفية الكتابة (textbackground ( ))

تستخدم لتغير لون خلفية الكتابة والصيغة العامة لها هي

textbackground (color no)

حيث يتم كتابة رقم اللون وهو من الرقم 1 الى الرقم 7

أو تكون الصيغة العامة لها كما يلى

textbackground (color name)

حيث يتم كتابة اسم اللون مثل اللون الاحمر (RED)

مثال صــ 200 هام جدا

تم بحمد الله

مع تمنياتى بالنجاح والتوفيق

محمد السيد إسماعيل

دوال الادخال و دوال الاخراج

1 – دالة الاخراج العامة (Printf ( ))

تستخدم فى اخراج البيانات العددية أو النصية أو السلاسل **والصيغة العامة لها هي :-**

اولا : البيانات العددية

وهى اما أن تكون متغيرات صحيحة أو متحركة أو ارقام

;printf ("format" , var1 , var2, ...)

اسم الدالة — توصيف الطباعة — المتغيرات

مثال

;printf("%d",x+y)

(حيث أن **y , x** متغيرات صحيحة **)**

ثانيا : البيانات النصية

وهى تشمل النصوص المراد اخراجها

;printf ("Text")

مثال

;printf("Mohammed:0124828233")

ثانيا : البيانات نصية وعددية معاً

حيث انه يمكن اخراج بيانات نصية وعددية معاً

مثل

;printf("x = %d y = %f " ,x,y)

وبفرض أن **x = 5 , y = 5.62**

اذاً يكون شكل الناتج كما يلى

x = 5 , y = 5.62

حيث انه اعتبر (**x** =) نص فيظهر كما هو ويقوم المترجة بطباعة قيمتة بعد علامة (=) بتوصيفة على انه صحيح (d%) وهكذا مع (**y**)

ملاحظات

- لابد أن يوضع التوصيف (d%) بين علامتى تنصيص مزدوجة (" التوصيف ")
- لابد من وصع علامة الفاصلة (,) قبل المتغير الاول و ايضا بين المتغيرات
- لابد ان ينتهى السطر بعلامة الفاصلة المنقوطة (;)
- يمكن أن يحتوى الجزء الخاص بتوصيف الطباعة (format) الموجود داخل دالة الاخراج على تسلسلات الهروب **مثل** printf("\n %d",x)

حيث تقوم سلسلة الهروب (n\ ) بطباعة الناتج على سطر جديد

بعض تسلاسلات الهروب

تعنى الانتقال الى سطر جديد	n\
تعنى اصدار صوت جرس	a\
تعنى البدء من اول السطر	r\
تعنى تقدم 7 مسافات عمودية قبل الطباعة	t\

امثلة عملية على دالة الاخراج printf

اعرض شكل الناتج الذى سيتم طباعتة من البرامج التالية

مثال 1

```
;Int c=3
;c =++c+c
;printf("The Value Of  c = %d",c)
```

ناتج الطباعة

The Value Of c = 8

مثال 2

```
;Int c=3
;printf("The Value Of  c = %d",++c+c)
```

ناتج الطباعة

The Value Of c = 8

نلاحظ ان المثالين لهم نفس الناتج ولكن الاختلاف هو ان العملية الحسابية فى البرنامج الاول فى سطر داخل البرنامج اما العملية الحسابية فى البرنامج الثانى تكون داخل دالة printf

مثال 3

```
;'Char x= 'a
;printf("\n x= %c",x)
```

ناتج الطباعة

x= a

مثال 4

```
;printf("%d ",100+10)
;printf("%f",100.0+10)
```

ناتج الطباعة

110
110.000000

فى عملية الطباعة الاولى المتغير صحيح وتوصيفة ايضا صحيح (d%) فبالتالى يكون الناتج صحيح (110) وبعد طباعة السطر الاول يتم الانتقال الى سطر جديد حيث انه يوجد تسلسل هروب (n\ ) للانتقال الى سطر جديد ثم طباعة السطر الثانى

فى عملية الطباعة الثانية المتغيرات احدها صحيح والاخر متحرك وطبقا لقاعدة الترقية يكون الناتج متحرك والتوصيف ايضا متحرك (f%) ويتم اضافة 5 اصفار عشرية

مثال 5

```
;printf("\n%d\n",140.5*2)
;printf("%f\n",140.5*2)
;printf("%f",140*2)
```

ناتج الطباعة

0
281.000000
0.000000

نلاحظ ان

- فى عملية الطباعة الاولى تم توصيف المتغيرات على انها صحيحة (% d) ولكن هى حقيقية لذلك يكون ناتج الطباعة خطأ ويظهر هذا الخطأ فى شكل صفر صحيح (0)
  ويمكن معالجة هذا الخطأ بالصيغة التالية printf("\n%d\n",int ((140.5*2));
  ففى هذه الحالة يكون الناتج (281)
- فى عملية الطباعة الثانية تم توصيف المتغيرات على انها حقيقية (% f) وهى بالفعل حقيقية لذلك يكون ناتج الطباعة حقيقى (281.000000) ويجب مراعاة موصف عرض الحقل وموصف الارقام العشرية ان وجد **مثل** (3.2f%) فيصبح الناتج (281.00)
- فى عملية الطباعة الثالثة تم توصيف المتغيرات على انها حقيقية (% f) ولكن هى صحيحة لذلك يكون ناتج الطباعة خطأ ويظهر هذا الخطأ فى شكل صفر متحرك

(0.000000) مثال صـ 176 (هام جدا)

---

2 – دالة الادخال العامة (scanf ( ))

تستخدم فى ادخال البيانات العددية أو النصية أو السلاسل **والصيغة العامة لها هي :-**

;scanf ("format" ,&var1 ,&var2, ...)

مثال

;scanf("%d",&x)

(حيث أن **x** متغير صحيح **)**

ملاحظات

- لابد أن يوضع التوصيف (d%) بين علامتى تنصيص مزدوجة (" التوصيف ")
- لابد من وصع العلامة (,&) قبل المتغير الاول و ايضا بين المتغيرات
- لابد ان ينتهى السطر بعلامة الفاصلة المنقوطة (;)
- يمكن أن يحتوى الجزء الخاص بتوصيف الطباعة (format) الموجود داخل دالة الاخراج على تسلسلات الهروب **مثل** scanf("\n %d",&x)
- اسماء المنظومات الحرفية يجب ألا يبدأ بالرمز (&) مثل scanf("%s",name)

امثلة عملية على دالة الادخال scanf

اعرض شكل الناتج الذى سيتم طباعتة من البرامج التالية

مثال 1

```
; Int I
;char a,b
;scanf("%c %c %d \n" ,&a ,&b ,&I)
;printf(" A = %c B = %c  I = %d " ,a ,b ,I)
```

اولا: يطلب البرنامج ادخال قيم للمتغير a , b , I واذا تم ادخال (mg1)

ثانيا : يظهر الناتج A= m B =g I = 1

مثال 2

```
;float a
;scanf("%f " ,&a )
;printf(" a = %.2f " ,a)
```

الناتج

اولا: يطلب البرنامج ادخال قيم للمتغير a واذا تم ادخال (2501.268533)

ثانيا : يظهر الناتج (2501.27)

مثال 3

(هام جدا) صــ 183

---

3 – دوال الادخال الخاصة (getchar ( ) , getche ( ) , getch ( ))

تستخدم جميعها لادخال حرف واحد فقط من لوحة المفاتيح

اولا : دالة getch ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح دون الضغط على زر Enter من لوحة المفاتيح وهذا الحرف لايظهر على الشاشة عند ادخاله

ثانيا : دالة getche ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح دون الضغط على زر Enter من لوحة المفاتيح وهذا الحرف يظهر على الشاشة عند ادخاله

ثالثا : دالة getchar ( )

وهى تستخدم لادخال حرف واحد للبرنامج من لوحة المفاتيح ولا يتم الانتقال الى الامر التالى إلا اذا تم الضغط على زر Enter من لوحة المفاتيح وهذا الحرف يظهر على الشاشة عند ادخاله

مثال 1

```
char a;
a = getche ( );
printf(" %c" ,a);
```

الدالة getche تودى الى ظهور الحرف a دون الضغط على enter لذلك فيظهر الحرف مرتين

4 – دوال الاخراج الخاصة (putch ( ))

تستخدم لادخال حرف فقط على الشاشة

```
char ch1, ch2, ch3;
printf ("\n ch1= ");
ch1=getchar ( );
```

```
;printf ("\n ch2= ")
;( ) ch2=getche
;printf ("\n ch3= ")
;( ) ch3=getch
        ;putch (ch1);  putch (ch2);    putch (ch3)
```

- فى السطر الثانى يطبع ch1= على سطر جديد
- فى السطر الثالث يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف m فيظهر هذا الحرف ولابد من الضغط على مفتاح enter للانتقال الى السطر التالى
- فى السطر الرابع يطبع ch2= على سطر جديد
- فى السطر الخامس يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف g فيظهر هذا الحرف ولا يتم الضغط على مفتاح enter حيث ينتقل تلقائيا الى السطر التالى
- فى السطر السادس يطبع ch3= على سطر جديد
- فى السطر السابع يقوم المتسخدم بإدخال حرف من لوحة المفاتيح وليكن الحرف s دون أن يظهر هذا الحرف ولا يتم الضغط على مفتاح enter حيث ينتقل تلقائيا الى السطر التالى
- فى السطر الثامن يطبع (m g s) **مثال صــ 188**

---

5 – دالة الاخراج (puts ( ))

تستخدم لطباعة عبارة أو منظومة حرفية فى نفس صورة ادخالها بدلاً من printf بالنسبة للمنظومات الحرفية

6 – دالة الادخال (gets ( ))

تستخدم لادخال عبارة أو منظومة حرفية فى نفس صورة ادخالها بدلاً من scanf بالنسبة للمنظومات الحرفية

ملاحظات

- دالة puts ( ) ودالة gets ( ) تستخدم للمنظومات والعبارات الحرفية فقط
- ودالة gets ( ) تستقيل البيانات الحرفية بكافة انواعها بما فيها الفراغات **مثل** عند ادخال كلمة (Mohammed Ismeal) تطبع كما هي بالفراغات التي بداخلها
- دالة scanf( ) لاتسمح بوجود مسافات داخل العبارة وتعتبرها فاصلة وتأخذ اول مقطع فقط

مثل عند ادخال كلمة (Mohammed Ismeal) تطبع المقطع الاول فقط Mohammed

مثال 1

```
char line [200];
gets (line);
puts (line);
```

الناتج

- فى السطر الثانى يطلب البرنامج ادخال منظومة حرفية تسمىline ومكونة من200 حرف فاذا قام المستخدم بأدخال

(When we take ..we may get little.. but when God gives... HE gives us more beyond our expectations)

- فى السطر الثالث يقوم البرانامج بطباعة العبارة فى نفس صورة ادخالها

7 – دالة مسح الشاشة (clrscr ( ))

تستخدم لمسح الشاشة ووضع المؤشر فى اول الركن العلوى الايسر من الشاشة

8 – دالة تغير لون الكتابة (textcolor ( ))

تستخدم لتغير لون الكتابة والصيغة العامة لها هي

textcolor (color no)

حيث يتم كتابة رقم اللون وهو من الرقم 0 الى الرقم 14

أو تكون الصيغة العامة لها كما يلى

textcolor (color name)

حيث يتم كتابة اسم اللون مثل اللون الاحمر (RED)

ملاحظات

- لطباعة نص باللون المحدد تستبدل دالة printf بدالة cprintf
- لطباعة اكثر من سطر كل سطر بلون (مثل حلقة) تستبدل سلسلة الهروب \n بـ\n\r

8 – دالة تغير لون خلفية الكتابة (textbackground ( ))

تستخدم لتغير لون خلفية الكتابة والصيغة العامة لها هي

textbackground (color no)

حيث يتم كتابة رقم اللون وهو من الرقم 1 الى الرقم 7

أو تكون الصيغة العامة لها كما يلى

textbackground (color name)

حيث يتم كتابة اسم اللون مثل اللون الاحمر (RED)

مثال صــ 200 هام جدا

تم بحمد الله

مع تمنياتى بالنجاح والتوفيق

محمد السيد إسماعيل